Inmagic content server

Web Services Guide

Inmagic Proprietary and Confidential

Copyright © 2004–2005 by Inmagic, Inc. All rights reserved.

Inmagic®, the Inmagic logo, DB/Text®, DB/TextWorks®, BiblioTech®, and BiblioTech PRO® are registered trademarks, and Inmagic.net™, BibSpeed™, IntelliMagic™, and PowerPack™ are trademarks of Inmagic, Inc. Other brand and product names are trademarks or registered trademarks of their respective holders.

The information in this document is subject to change without notice and should not be construed as a commitment by Inmagic, Inc., which assumes no responsibility for any errors that may appear in this document. Use of any other product name does not imply endorsement of that product by Inmagic, Inc.

This documentation may be used only in accordance with the terms of the Inmagic license agreement.

WARRANTY

INMAGIC, INC. MAKES NO WARRANTY, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY AND FITNESS. INMAGIC, INC. SHALL NOT BE LIABLE FOR ANY LOST PROFITS OR ANY OTHER INCIDENTAL OR CONSEQUENTIAL DAMAGES. IN PARTICULAR, INMAGIC, INC. SHALL HAVE NO LIABILITY FOR ANY DATA OR PROGRAMS STORED OR USED WITH THIS PRODUCT, INCLUDING THE COSTS OF RECOVERING SUCH PROGRAMS OR DATA.

U.S. GOVERNMENT: If Licensee is acquiring the software on behalf of any unit or agency of the U.S. Government, the following shall apply:

(a) For units of the Department of Defense: RESTRICTED RIGHTS LEGEND: Use, duplication or disclosure by the Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data Clause at DFARS 252.227-7013. (b) For any other unit or agency: NOTICE - Notwithstanding any other lease or license agreement that may pertain to, or accompanying the delivery of, the computer software and accompanying documentation, the rights of the Government regarding its use, reproduction and disclosure are as set forth in Clause 52.227-19(c)(2) of the FAR.

Contractor/Manufacturer is Inmagic, Inc., 200 Unicorn Park Drive, Fourth Floor, Woburn, MA 01801, U.S.A.

Questions regarding any Inmagic product should be addressed to Inmagic, Inc., or your local authorized Inmagic dealer.

Inmagic, Inc.
200 Unicorn Park Drive
Fourth Floor
Woburn, MA 01801 U.S.A.
Telephone: 781-938-4444 or 800-229-8398
Fax: 781-938-4446
http://www.inmagic.com

r0505/CS13

Contents

Preface ..... **v**
Other Documentation v
Contacting Inmagic, Inc. vi
Web Services – SOAP Interface **1**
How to Create a SOAP Interface for a Textbase **2**
The SOAP Interface **3**
Operations (as seen in Visual Studio .NET 2003) 4
Struct model 4
query 4
insert 4
update 5
delete 5
qdelete 5
Dataset model 6
queryDS 6
insertDS 6
updateDS 6
deleteDS 7
qdeleteDS 7
XML model 8
queryX 8
insertX 8
updateX 8
deleteX 9
qdeleteX 9
File List 10
Error Handling 10
How to Use the SOAP Interface **11**
Using the SOAP Interface with Visual Studio .NET 2003 11
Textbase Design Hints 12
Index **13**

Preface

This technical guide is intended for developers, system integrators, and partners of Inmagic, Inc. who will be using the Inmagic® Web services - SOAP interface to develop their own applications for use with Inmagic Content Server. Note that the Inmagic Web Services - SOAP interface is suited for the Microsoft Visual Studio .NET 2003 development environment, in which it was tested. It has not been validated in any other development environments.

Other Documentation

The following documentation is also available for *Content Server*:

- ***Inmagic Content Server User's Manual*.** A user's manual that covers CS/TextWorks and CS/*WebPublisher PRO* and is available as a PDF file, which is included with the installation of the software.
- **Online help.** The help file, which covers CS/TextWorks, CS/*WebPublisher PRO*, and CS/*Importer*, has the most up-to-date information and includes more detail than the user's manual. To open the help file:
  - Start CS/TextWorks and press **F1**, or choose **Help>Help Topics** from the menu bar. For context-sensitive help, click on the toolbar, then click the menu command or toolbar button that you want to know about. For a help topic on specific tasks you can perform in a window, click the Help on this Window button .
  - Start CS/*Importer* and press the **Help** button.
- **README file.** This file tells you what is new with that particular version of the software; lists issues resolved from the previous release and any late-breaking information; and contains other important information. The README file is installed in the CS/TextWorks and CS/*WebPublisher PRO* installation folders. Open it using a Web browser.
- **Installation Notes.** This booklet, which comes with the software, explains how to install the software. It is also available as a PDF file.
- **Knowledgebase and Web site.** Provides information via the Web. With the software open and a Web browser available:
  - To search for solutions to common problems, choose **Help>Inmagic on the Web>Knowledgebase** to go to the Inmagic Product Support knowledgebase.
  - To visit the Inmagic Web site, choose **Help>Inmagic on the Web>Home Page** or **Help>Inmagic on the Web>Support Page**.
- **Inmagic Web Products Schema.** This file contains information about each element you can submit in XML to CS/*WebPublisher PRO*. It is located on the Inmagic Web site at http://support.inmagic.com/web.
- **HTML help pages for CS/*WebPublisher PRO*.** Provides help topics for the Web component. See the online help for descriptions of WEB_BEGIN.HTM, WEB_ICHOICES.HTM, WEB_MSG.HTM, and WEB_WW.HTM, which are in the HELP subfolder of the folder into which you installed CS/*WebPublisher PRO*.

- **Administrator's Guide.** This guide describes ongoing administrative and maintenance tasks for the *Content Server* system that should be performed by the Content Server Administrator (for example, how to back up and restore a *Content Server* textbase). Available as a PDF file, it is located on the Inmagic Web site at http://support.inmagic.com/web.
- ***Upgrading to the Inmagic Content Server Platform.*** This document explains how to upgrade from Inmagic DB/TextWorks® and Inmagic DB/Text *WebPublisher PRO* to CS/TextWorks and CS/*WebPublisher PRO*. Available as a PDF file, it is installed with your *Content Server* software and is available on the Inmagic Web site at http://support.inmagic.com/web.

Contacting Inmagic, Inc.

If you have tried all the resources listed above and you still need help, you can contact Inmagic, Inc. or your local Inmagic dealer.

If you have a Technical Support contract, please have the number handy, and try to be at your computer when you call. If that is not possible, note exactly what you were doing when you encountered the problem, the exact text of any error messages you received, and your software version and serial number (choose **Help>About CS/TextWorks** to look it up). If you do not have a Technical Support contract, you can contact Inmagic Sales to purchase a support plan.

Inmagic, Inc.
200 Unicorn Park Drive, Fourth Floor
Woburn, MA 01801, U.S.A.
Tel: 781-938-4444 or 800-229-8398
Fax: 781-938-4446
http://www.inmagic.com

support@inmagic.com	- technical support questions
CustomerSvc@inmagic.com	- general company, product, and services questions
sales@inmagic.com	- sales, product pricing, and custom solution questions
wishlist@inmagic.com	- feature requests

If your message is intended for a particular person at Inmagic, Inc. (for example, a Product Support representative who is expecting the message or files), please include the name of that person in the subject and the message.

Communicating with Other Users

You can participate in user-to-user discussions through an Inmagic forum on the Web. Note that the forums are not an official customer support channel for Inmagic products. To participate in a forum, go to the Inmagic Web site (http://www.inmagic.com) and from the **Support** menu, click **Forums**.

Web Services – SOAP Interface

This document describes the Web services - SOAP interface available for use with the Inmagic® *Content Server* product line. Note that the Inmagic Web Services - SOAP interface is suited for the Microsoft Visual Studio .NET 2003 development environment, in which it was tested. It has not been validated in any other development environments.

Inmagic *Content Server* is an enterprise-wide scalable content management system. It combines the advantages of a robust and flexible database management environment with high speed search and categorization, making finding relevant information fast, easy, and precise. Web publishing capabilities are built into Inmagic *Content Server*, offering a better way to publish, access, and maintain information on corporate intranets and the Internet.

Content Server provides you with access to Inmagic.net™, which is what you use to create a Web Services Description Language (WSDL) file needed to create a SOAP interface. Inmagic.net also gives you the ability to connect to the Inmagic server from within a CS/TextWorks session to accomplish various tasks relating to acquiring, cataloging, indexing, and disseminating information throughout an organization.

Content Server is built on industry standard architecture, incorporating Microsoft SQL Server 2000 and/or Microsoft SQL Server Desktop Engine (depending on the configuration of *Content Server* you have) as its data store. *Content Server* makes extensive use of open standards and technologies, the most obvious being Extensible Markup Language (XML), which forms the basis for much of its Web publishing functionality.

Using the components of *Content Server*, you can display, add, modify, and delete information in a textbase from your desktop and/or through a Web browser.

Content Server is made up of three components:

- CS/TextWorks. This is the desktop component.
- CS/*WebPublisher PRO*. This is the component that enables you to publish your textbases on the Web. Anyone with a Web browser can point to an HTML screen and submit a query. The results are returned dynamically, in forms that you design. You can also design an interface so you and/or your users can add, edit, and delete records in a textbase through the Web.
- CS/*Importer*. This component is a service that enables you to load records into one or more textbases, while automatically and continually processing imports in the background. You can import records in Inmagic tagged, delimited ASCII, and XML formats. You can also import documents (for example, Microsoft Word documents).

The Standard and Enterprise editions of *Content Server* include both components, while the Workgroup edition can be purchased with both or just the desktop component.

Inmagic Proprietary and Confidential

How to Create a SOAP Interface for a Textbase

You can create a textbase-specific WSDL file and accompanying transform and schema files using Inmagic.net.

To created a SOAP interface for a textbase

1. Start CS/TextWorks and open the textbase for which you want to create a SOAP interface.
2. Choose **Inmagic.net>Configure**.
3. Click the **Create SOAP Interface** button.
4. In the first box, choose a field that uniquely identifies a record in a textbase. The field must be term-indexed, present in every record, and unique in every record. An Automatic Number field would be ideal.
5. In the second box, type the name of the server on which the SOAP interface will be installed.
6. Click the **Create SOAP Interface** button. A message appears when the files have been created, including the location and names of the files. The location is wherever the textbase resides.
7. Close the software and copy these files to the SOAP subfolder of the CS/*WebPublisher PRO* installation folder. The following files are created (your textbase name appears where <textbase> is shown; for example, cars.wsdl):
   - <textbase>.wsdl
   - <textbase>_input.icx
   - <textbase>_output.xsl
   - <textbase>_outputDS.xsl
   - <textbase>_outputX.xsl
   - <textbase>_recordset.xsd
   - <textbase>_result.xsd

Inmagic Proprietary and Confidential

The SOAP Interface

There are five (5) basic functions supported in the SOAP interface: query (with optional sorting), insert, update, delete, and qdelete. There are three (3) data models available for each of these functions: Struct, Dataset, and XML. The operations that implement these functions are listed in the following table:

Struct	Dataset	XML
query	queryDS	queryX
insert	insertDS	insertX
update	updateDS	updateX
delete	deleteDS	deleteX
qdelete	qdeleteDS	qdeleteX

In the **Struct model**, records are represented as nested arrays of strings. A set of records is an array of records, where a single record is a struct or class whose elements are field names. A field is either a string (in the case of a field that cannot repeat) or an array of strings (in the case of a field that can repeat). A field is assumed to be repeating unless it is an Automatic or Computed field, or it has single-entry-only validation applied.

In the **Dataset model**, a set of records is represented in an ADO.NET dataset. A "record" table in the dataset contains columns for all of the nonrepeating fields. Repeating fields each have a table of their own, automatically related to the record table. The Dataset model is especially suited to the Microsoft Visual Studio .NET 2003 development environment.

In the **XML model**, record sets, records, and fields are all XML elements, nested as you would expect. The field names serve as the XML tags identifying the content.

Inmagic Proprietary and Confidential

Operations (as seen in Visual Studio .NET 2003)

This section lists each operation, as seen in Visual Studio .NET 2003, and its parameters and returns. In Visual Studio .NET 2003, all parameters are required. Therefore, you must use an explicit null in place of any parameter that you do not want to use.

Struct model

The Struct model has the following operations, each with its own parameters and returns.

query

Parameters:

string – query string in command-query format (for example, `Find Author CT Smith`).

string[] – array containing names of fields to return. If null, all fields are returned.

string[] – array containing names of fields to sort result on, most significant first. If a name is prefixed by "-" (minus sign or hyphen), a descending sort is used. If null, the textbase default sort is used.

string – password with which to open the textbase. If null, no password is passed in.

Returns:

<textbase>_recordsettypeRecord[] – array of records retrieved, where each record consists of a struct with a string member for each nonrepeating field and an array-of-strings member for each repeating field.

---

insert

Parameters:

<textbase>_recordsettypeRecord[] – array of new records to be inserted into the textbase.

string[] – array containing the names of log fields to be returned in the confirmation response.

string – password with which to open the textbase. If null, no password is passed in.

Returns:

<textbase>_resulttype – log fields and status (succeeded/failed) for each record submitted; also contains error information.

Inmagic Proprietary and Confidential

update

Parameters:

<textbase>_recordsettypeRecord[] – array of records to be modified in the textbase.

string[] – array containing the names of log fields to be returned in the confirmation response.

string – password with which to open the textbase. If null, no password is passed in.

Returns:

<textbase>_resulttype – log fields and status (succeeded/failed/amended) for each record submitted; also contains error information.

---

delete

Parameters:

<textbase>_recordsettypeRecord[] – array of records to be deleted from the textbase.

string[] – array containing the names of log fields to be returned in the confirmation response.

string – password with which to open the textbase. If null, no password is passed in.

Returns:

<textbase>_resulttype – log fields and status (succeeded/failed) for each record submitted; also contains error information.

---

qdelete

Parameters:

string – query string in command-query format (for example, `Find Author CT Smith`); all records found by this query will be deleted.

string[] – array containing the names of log fields to be returned in the confirmation response.

string – password with which to open the textbase. If null, no password is passed in.

Returns:

<textbase>_resulttype – log fields and status (succeeded/failed) for each record retrieved by the query; also contains error information.

Dataset model

The Dataset model has the following operations, each with its own parameters and returns.

queryDS

Parameters:

string – query string in command-query format (for example, `Find Author CT Smith`).

string[] – array containing names of fields to return. If null, all fields are returned.

string[] – array containing names of fields to sort result on, most significant first. If a name is prefixed by "-" (minus sign or hyphen), a descending sort is used. If null, the textbase default sort is used.

string – password with which to open the textbase. If null, no password is passed in.

Returns:

<textbase>_recordset – dataset containing the records retrieved.

---

insertDS

Parameters:

<textbase>_recordset – dataset containing the new records to be inserted into the textbase. If any records in the dataset have changes of type "inserted," only those records are added to the textbase. If no records have changes, all are added to the textbase.

string[] – array containing the names of log fields to be returned in the confirmation response.

string – password with which to open the textbase. If null, no password is passed in.

Returns:

<textbase>_result – dataset containing result of the operation.

---

updateDS

Parameters:

<textbase>_recordset – dataset containing the records to be modified in the textbase. If any records in the dataset have changes of type "inserted" or "modified," only those records are updated or added. If no records have changes, all are updated or added.

string[] – array containing the names of log fields to be returned in the confirmation response.

string – password with which to open the textbase. If null, no password is passed in.

Returns:

<textbase>_result – dataset containing result of the operation.

deleteDS

Parameters:

<textbase>_recordset – dataset containing the records to be deleted from the textbase. If the dataset contains deleted records, only those records are deleted from the textbase. If the dataset contains no deletions, all of the records are deleted from the textbase.

string[] – array containing the names of log fields to be returned in the confirmation response.

string – password with which to open the textbase. If null, no password is passed in.

Returns:

<textbase>_result – dataset containing result of the operation.

---

qdeleteDS

Parameters:

string – query string in command-query format (for example, `Find Author CT Smith`); all records found by this query will be deleted.

string[] – array containing the names of log fields to be returned in the confirmation response.

string – password with which to open the textbase. If null, no password is passed in.

Returns:

<textbase>_result – dataset containing result of the operation.

XML model

The XML model has the following operations, each with its own parameters and returns.

queryX

Parameters:

string – query string in command-query format (for example, `Find Author CT Smith`).

string[] – array containing names of fields to return. If null, all fields are returned.

string[] – array containing names of fields to sort result on, most significant first. If a name is prefixed by "-" (minus sign or hyphen), a descending sort is used. If null, the textbase default sort is used.

string – password with which to open the textbase. If null, no password is passed in.

Returns:

System.Xml.XmlElement – XML structure conforming to the <textbase>_recordset schema.

---

insertX

Parameters:

System.Xml.XmlNode – XML structure conforming to a <textbase>_record schema containing new records to be inserted into the textbase.

string[] – array containing the names of log fields to be returned in the confirmation response.

string – password with which to open the textbase. If null, no password is passed in.

Returns:

System.Xml.XmlNode – XML structure conforming to the <textbase>_result schema.

---

updateX

Parameters:

System.Xml.XmlNode – XML structure conforming to a <textbase>_record schema containing records to be updated in the textbase.

string[] – array containing the names of log fields to be returned in the confirmation response.

string – password with which to open the textbase. If null, no password is passed in.

Returns:

System.Xml.XmlNode – XML structure conforming to the <textbase>_result schema.

deleteX

Parameters:

System.Xml.XmlNode – XML structure conforming to a <textbase>_record schema containing records to be deleted from the textbase.

string[] – array containing the names of log fields to be returned in the confirmation response.

string – password with which to open the textbase. If null, no password is passed in.

Returns:

System.Xml.XmlNode – XML structure conforming to the <textbase>_result schema.

---

qdeleteX

Parameters:

string – query string in command-query format (for example, `Find Author CT Smith`); all records found by this query will be deleted.

string[] – array containing the names of log fields to be returned in the confirmation response.

string – password with which to open the textbase. If null, no password is passed in.

Returns:

System.Xml.XmlNode – XML structure conforming to the <textbase>_result schema.

File List

This section lists the files that comprise the SOAP interface for one textbase and describes their functions.

File	Description
<textbase>.wsdl	Web Services Description Language file, containing a textbase-aware specification of the SOAP operations described in the previous section.
<textbase>_input.icx	Input transform, used to convert an incoming SOAP message to XML suitable for submission directly to CS/*WebPublisher PRO*. This file has an ICX extension because that extension has been mapped to CS/*WebPublisher PRO*.
<textbase>_output.xsl	Output transform, used to convert XML returned by CS/*WebPublisher PRO* into a SOAP response, using the Struct model.
<textbase>_outputDS.xsl	Output transform, used to convert XML returned by CS/*WebPublisher PRO* into a SOAP response, using the Dataset model.
<textbase>_outputX.xsl	Output transform, used to convert XML returned by CS/*WebPublisher PRO* into a SOAP response, using the XML model.
<textbase>_recordset.xsd	Schema describing the recordset returned after a query operation.
<textbase>_result.xsd	Schema describing the result (logfields and status) returned after an insert, update, or delete operation. Also describes an error returned, if applicable.

Error Handling

Errors may be returned as SOAP faults or as information contained in the response. SOAP faults are generated when the software is unable to process the request. Some examples of problems that would generate a SOAP fault include query syntax errors, textbase access problems, or operations involving textbase fields that were deleted or renamed after the SOAP interface was created. Some examples of errors that would be returned in the response include an attempt to delete a record that does not exist, or enter invalid information in a field.

How to Use the SOAP Interface

This section describes how to use the SOAP interface with Visual Studio .NET 2003.

Using the SOAP Interface with Visual Studio .NET 2003

After you have created the SOAP interface and copied the files to your CS/*WebPublisher PRO* SOAP subfolder, create a Visual Studio .NET 2003 project and add a Web reference. Navigate to the SOAP subfolder and type the **<textbase>.WSDL** filename (for example, `http://<hostname>/ics-wpd/SOAP/cars.wsdl`). Visual Studio .NET 2003 will generate a namespace with the name you selected for the Web reference. This namespace contains the following classes, where <textbase> represents your textbase name (for example, `Cars_recordset` if your textbase name is Cars).

Class	Description
<textbase>_recordset	Dataset containing records (Dataset model)
<textbase>_recordsettypeRecord	struct containing records (Struct model)
<textbase>_result	dataset containing result of an update, insert, or delete operation (Dataset model)
<textbase>_resulttype	struct containing result of an update, insert, or delete operation (Struct model)
<textbase>_resulttypeRecord	struct containing record information after an update, insert, or delete (log fields, status, and error information)
<textbase>_resulttypeSummary	struct containing summary information after an update, insert, or delete (number of operations that succeeded, failed, and so forth)
<textbase>_SOAP	class containing the operations (query, queryDS, and so forth) as methods

Tip! If you decide to use the Dataset model (recommended in Visual Studio .NET 2003), after you add the Web reference, add the dataset classes to your Web form. You can then bind them to the control(s) of your choice (for example, a DataGrid).

Textbase Design Hints

To maximize the ease of building your Web application using the SOAP interface, review these recommendations:

- Make sure your textbase contains a field that is unique and required, to serve as a record identifier.
- Consider applying single-entry-only validation to fields that you would never expect to repeat (contain multiple entries). In the Dataset model, nonrepeating fields are returned in one table; repeating fields each have a table of their own. A field is assumed to be repeating unless it is an Automatic or Computed field, or it has single-entry-only validation applied.
- Ensure that your textbase does not contain multiple fields whose names would be equivalent when they are made SOAP-compliant. For example, `First Name` and `First-Name` both appear as `First_Name` in the SOAP interface. See the *Content Server* online help for information about SOAP-compliant field names.

(Note that the phrase "SOAP-compliant" is a bit misleading. SOAP itself has no problem with hyphens, but Microsoft Visual Studio .NET 2003 does not handle them well in variable names. The SOAP interface automatically uses SOAPFormat=1; therefore there is no need to specify it separately in the ICSWEB.INI file.)

Index

A

About CS/TextWorks command, vi

C

Content Server, 1
context-sensitive help, v
CS/*Importer*, 1
CS/TextWorks, 1
CS/*WebPublisher PRO*, 1
customer support, vi

D

data models, 3
Dataset model, 3, 6
delete, 3, 5
deleteDS, 7
deleteX, 9
documentation, v

E

error handling, 10

F

file list, 10
forums, vi
functions, 3

H

help, v
hints for textbase design, 12

I

ICSWEB.INI, 12
Inmagic, Inc., vi
insert, 3, 4
insertDS, 6
insertX, 8
installation notes, v
Internet, v

K

knowledgebase, v

M

Microsoft Visual Studio .NET, 3, 4, 11

N

namespace, 11

O

online help, v
operations, 4

P

Product Support, vi

Q

qdelete, 3, 5
qdeleteDS, 7
qdeleteX, 9
query, 3, 4
query syntax error, 10
queryDS, 6
queryX, 8

R

README file, v

S

serial number, vi
SOAP fault, 10
SOAP interface, 3
SOAP-compliant, 12
Struct model, 3, 4
support, vi
syntax error, 10

T

technical support, vi
textbase design hints, 12
troubleshooting, 10

U

update, 3, 5
updateDS, 6

updateX, 8
user's manual, v

W

Web forums, vi
Web Services Description Language file, 1, 2, 10
Web site, v
WSDL file, 1, 2, 10

X

XML model, 3, 8